Goanna SSS Viewer 0.3

Michael Ekstrand
Research Assistant
Ames Lab - Scalable Computing Laboratory

331 Wilhelm Hall
Iowa State University
Ames, IA 50011
mekstran@scl.ameslab.gov

Goanna SSS Viewer 0.3

by Michael Ekstrand

Table of Contents

List of Tables

Chapter 1. Introduction

Goanna is a desktop program for viewing and interacting with cluster monitor data returned by *Scalable Systems Software (SSS)* components. It allows the user to view node status and performance, visualize network topography, and aids in debugging cluster and network difficulties.

Chapter 2. Goanna Concepts

Terminology

Goanna uses a few specific terms throughout its interface (and this manual). They are defined in the glossary, but some terms are introduced here to aid in understanding the remainder of the manual.

The fundamental entity in Goanna is the *system*. A system is a cluster, supercomputer, network, or other self-contained object that Goanna is being used to examine. Each Goanna data file is for one system.

Systems (usually) contain multiple nodes. A *node* is a computer, network infrastructure device (such as a switch), or other constituent component of a system. Most things in Goanna deal with nodes and their properties.

Goanna's view of a system

Nodes

Goanna considers a system to be comprised of nodes. Each node has various properties, some static and some dynamic. Each node may also have network devices, which in turn have network ports. Network ports are connected to network ports in other devices, frequently in other nodes.

A node's *static properties* are properties which are considered essentially immutable for the lifetime of the node. Things like system architecture, operating system, and installed processors and disk space are static properties. Goanna assumes that any change in these properties needs to be confirmed, and that data about the static configuration can be saved in the data file.

A node's *dynamic properties* are properties which change frequently in the course of system operation. Things such as available memory and CPU utilization fall into this category. Goanna does not save any dynamic properties in its saved data file.

Goanna also tracks the Infiniband networking capabilities of the nodes in the system. Each node has network devices, which have ports. Goanna saves the list of devices, ports, and remote targets for ports (the GUID and port number at the other end of a connection).

Topography

Goanna builds a view of the network topography, built on a logical topography derived from the connections between nodes. It augments this with data as to the 3-D position of each node in the visualization, and with a hierarchical container mechanism for managing visual layout. At present, the hierarchical system is little-used, but it will form the basis for future enhancements to the topography view.

Saved data files

Goanna has saved data files which store information about a cluster. It stores a descriptive name for the system, static information for saved nodes, some topography data, and information on what SSS servers to query.

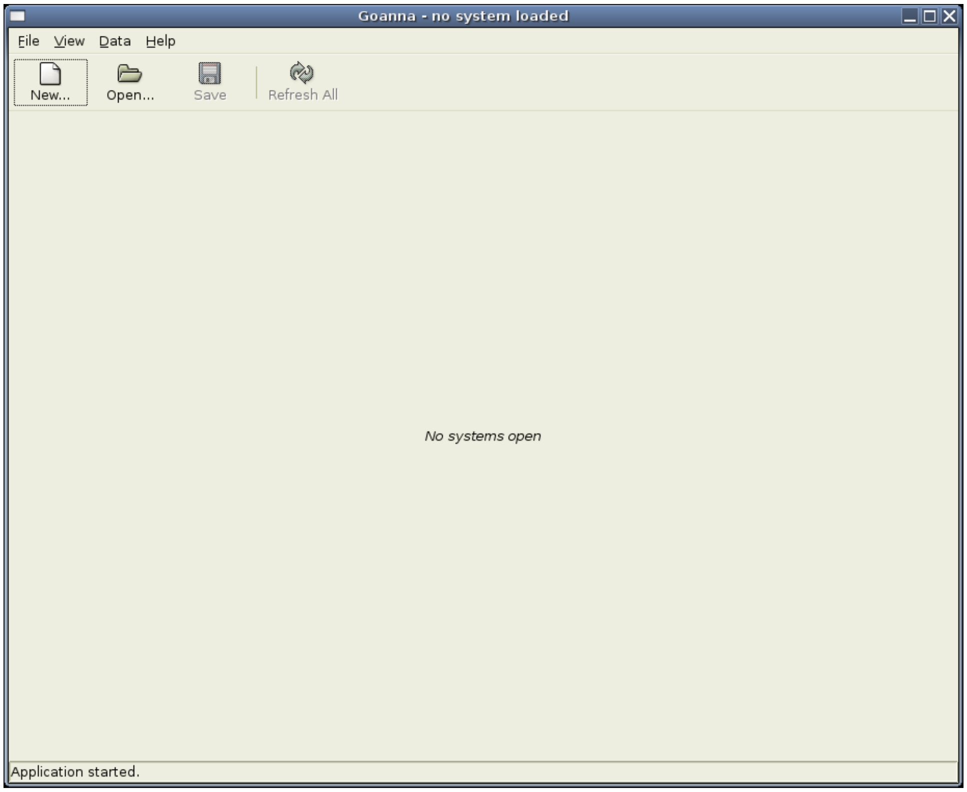
> **Note:** Goanna will *never* modify saved data without explicit instruction. If a server returns data which conflicts with data saved in the data file, that data will not be saved until the conflict is resolved.

> **File format compatibility:** Goanna attempts to keep its file formats forwards and backwards compatible; see Appendix D for details.

Nodes are not automatically saved in the data file; adding nodes to Goanna's saved state is discussed in Chapter 4.

Chapter 3. Interface Overview

Figure 3-1. Main Window

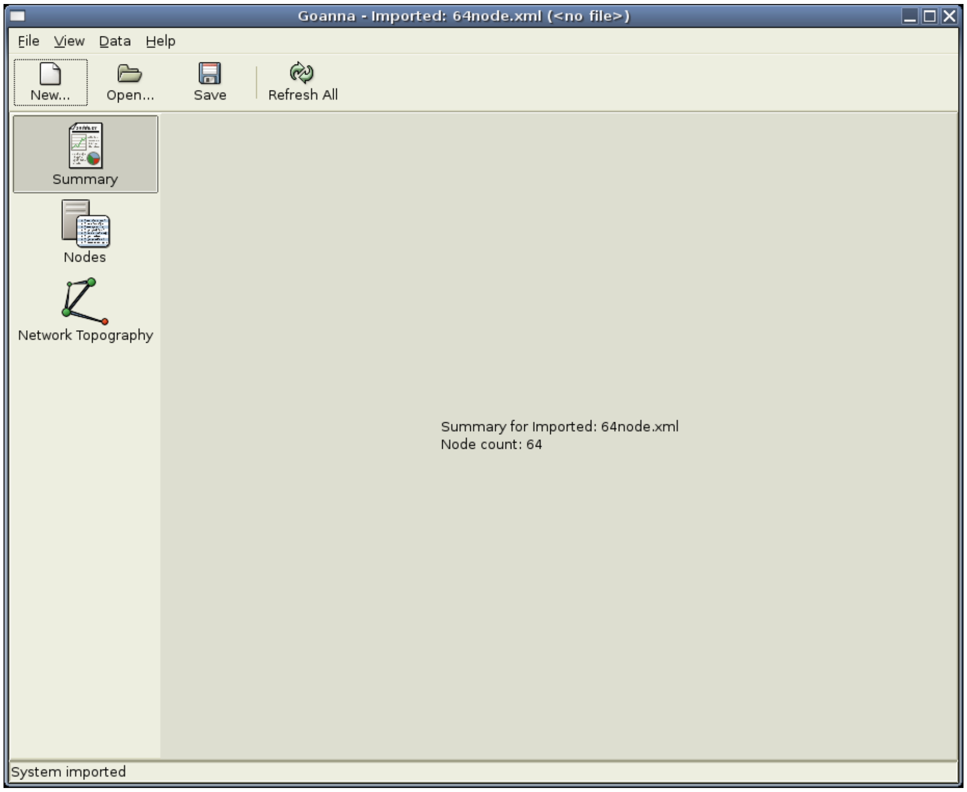


Goanna’s main application window with no system open.

When Goanna is first started, it presents the interface shown in Figure 3-1. From this window, a new system can be created or an existing system opened.

The System View

Figure 3-2. System View



Summary for Imported: 64node.xml
Node count: 64

System imported

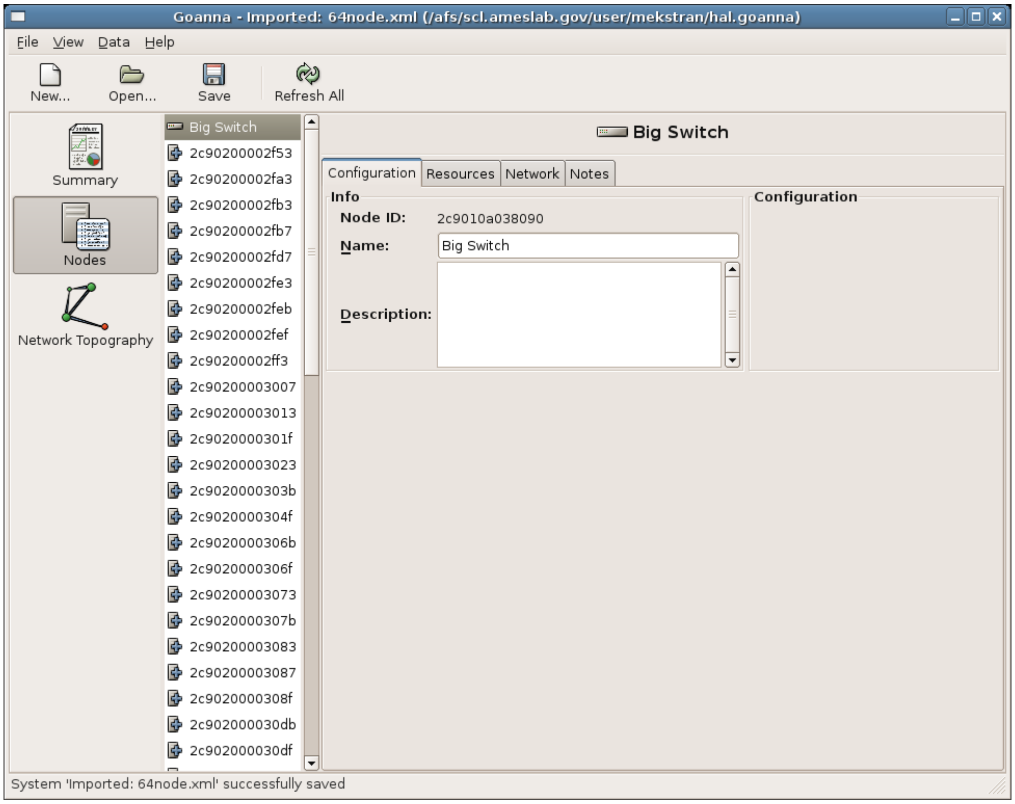
Initial view after opening a system

Once a system has been opened, Goanna presents the system view, as in Figure 3-2. This provides access to the different views available for a system (via the icons on the left side of the window), as well as the capacity to save and otherwise manipulate the system.

Chapter 4. The Node Viewer

The “Nodes” view provides a list of all the nodes in a system, and can be used to add nodes to the data file and view and manage the data available about the nodes, both saved and unsaved.

Figure 4-1. Node View



Big Switch
2c90200002f53
2c90200002fa3
2c90200002fb3
2c90200002fb7
2c90200002fd7
2c90200002fe3
2c90200002feb
2c90200002fef
2c90200002ff3
2c90200003007
2c90200003013
2c9020000301f
2c90200003023
2c9020000303b
2c9020000304f
2c9020000306b
2c9020000306f
2c90200003073
2c9020000307b
2c90200003083
2c90200003087
2c9020000308f
2c902000030db
2c902000030df

Node ID: 2c9010a038090

Name: Big Switch

Description:

System 'Imported: 64node.xml' successfully saved

The “Nodes” view.

Node Data Icons

Goanna uses icons to indicate what kind of node a particular node is, as well as its current data status.

Table 4-1. Node type icons

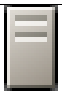
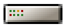
Icon	Description
	Node
	Network switch

Table 4-2. Data status emblems

Icon	Meaning
✚	The object is new (it has not yet been added to the data file).
◔	New data available. The object is saved in the data file, but there is new data available for this object which has not yet been saved (but which is consistent with the saved data).
❗	New data is available, but this data conflics with currently saved data (e.g., the node is saved as running Linux, and the node monitor reports AIX).
💾	The savable state for the object has changed, but these changes have not yet been saved to disk (this emblem may appear in conjunction with another emblem).

Saving Nodes

Before Goanna saves any data pertaining to a node, it is necessary to add the node to Goanna's list of saved noes. This is done with the Add (check mark) button in the node viewer. When that button is clicked, Goanna adds that node, and any self-consistent savable data pertaining to it, to the saved node set (to be written to the data file on the next save).

Goanna will automatically add any network devices that are a part of the node to the data file when the node is added. Network devices that show up later can be added separately, with the Add Device button.

> **Note:** Goanna does not yet provide a means to remove data from the data file. That functionality is on the TODO list, but a sensible UI for doing so has not yet been determined. Feedback on this is welcome; see the Section called *Project Status* in Appendix A.

Chapter 5. Topography Viewer

Topography Controls

The toolbar

The toolbar contains several items for controlling the topography view.

Table 5-1. Topography toolbar items

Icon	Description
Home Icon	“Home” the view. Zooms all the way out and moves the camera to a default position and orientation. This option will probably be reworked heavily in a future version.
	Toggle electrostatic repulsion. See the Section called *Electrostatic Repulsion*.

Positioning

The positioning model of the topography view determines where objects are drawn in three-dimensional space. Each object has a position, which is saved in the data file. That position can be fixed, so that Goanna will not move the object when it is automatically adjusting positions. A combination of fixed positions and automatic positioning (see the Section called *Electrostatic Repulsion* later in this chapter) allows the user to position key items in the topography, and then have the rest of the visualization position itself with respect to those key positions.

Electrostatic Repulsion

Goanna implements an electrostatic repulsion algorithm for determining reasonable positions of unpositioned objects. When electrostatic repulsion is enabled (see Table 5-1), all objects without fixed positions push on each other, in a similar manner to electrostatic charges. Links between objects pull those objects towards each other, so that the final result is a reasonably-distributed initial display.

Objects with fixed positions remain stationary when electrostatic repulsion is positioning objects; they do, however, continue to exert force on their neighbors.

Appendix A. Project Information

Goanna, being an open-source project, has a variety of online resources available for users and potential developers. This appendix provides information on those project resources, as well as other general project information.

Why call it Goanna?

Goanna is a monitor program. *Goanna* is also the name of the Australian monitor lizard. Monitor program, monitor lizard, it just might fit.

Project Status

Goanna is currently in a relatively early stage of development. It is useful for playing with the technology, and as an experimentation platform to see what types of features would be useful in a GUI monitor program. Feedback is greatly appreciated; we have some ideas as to what capabilities should be implemented on top of Goanna's framework, but real-world usage scenarios and feature request will help guide development. E-mail Michael Ekstrand at <`mekstran@scl.ameslab.gov`> with any ideas or suggestions, or put them in Trac as Enhancement tickets.

Web site

Goanna and its sibling projects are managed via a Trac instance (http://sss.scl.ameslab.gov/trac). This site contains further documentation, the bug tracker, source viewer, etc.

> **Note:** As of August, 2006, the CC functionality in Trac is not enabled. Therefore, specifying an e-mail address in the CC field in a Trac ticket will not result in e-mail notification of ticket changes at this time.

Support

What support there is presently consists of e-mailing the author, Michael Ekstrand, at <`mekstran@scl.ameslab.gov`>.

Reporting Bugs

Bug reports, feature requests, and patches can be submitted as tickets in the Goanna Trac instance. See the Section called *Web site* for more information on the Goanna web resources.

Appendix B. Advanced Invocation

Usually, Goanna will be started from its desktop or application menu icon. However, it can also be started using the **goanna** command in a terminal, and several facets of its behavior can be controlled in this manner.

Command Line Usage

Synopsis

goanna [`OPTION`...] [`FILE`...]

Options

Below is a summary of relevant command line options. For more detailed information, run **`goanna --help`**.

`-?`
`--help`

Show command line usage information

`--version`

Print version information and exit

`--log-level=LEVEL`

Specify a run-time logging level. Valid levels are:

- `fatal`
- `error`
- `warn`
- `info` (the default for non-debug builds)
- `debug`
- `trace`
- `mtrace`
- `ntrace`

`--disable-gl`

Disable all OpenGL support (disables the topography view module)

`--display=`*`DISPLAY`*

Specifies the X display on which Goanna is to run; overrides the DISPLAY environment variable.

`--gdk-gl-force-indirect`

Disables direct rendering in OpenGL

> **Note:** This option does not appear in the output of `--help` for technical reasons, but is useful for getting Goanna to work on some graphics hardware.

Additional Information

Goanna is a GTK+ application using GtkGLExt. As such, it takes all the standard GTK and GtkGLExt options. See *Running GTK+ Applications* (http://developer.gnome.org/doc/API/2.0/gtk/gtk-running.html) and *Running GtkGLExt Applications* (http://gtkglext.sourceforge.net/reference/gtkglext/gtkglext-running.html) for more information on the options and environment variables that control those libraries.

Appendix C. Troubleshooting (Help! It doesn't work!)

> **Note:** Check the *Goanna Issues* page (http://sss.scl.ameslab.gov/trac/wiki/GoannaIssues) for up-to-date information on known Goanna (or surrounding environment) issues and workarounds.

Topography view issues

Replicating with indirect rendering

Some problems with Goanna's topography view are caused by graphics driver issues. Forcing Goanna to use indirect rendering for its visualization can get around some problems (such as the lack of GL selection support in the r300 driver).

To disable direct rendering in Goanna, either use the `--gdk-gl-force-indirect` option or set the GDK_GL_FORCE_INDIRECT environment variable to `1` when running Goanna.

Appendix D. Compatibility Notes

File format compatibility

Every effort is made to maintain the comapatibility of the Goanna file format. Files created by a particular version of Goanna should be readable in a previous version, without losing any data upon resave that was unknown to the earlier version, and later versions should always be able to load a previous version's files.

That said, there are times when it is necessary to break compatibility. Major breaks will result in new XML namespaces. This section will attempt to document any incompatibilities.

Change in file source configuration

In Goanna version 0.3, file source filenames were stored in a `filename` element. It was later discovered that this caused some filename encoding consistency problems which may cause problems on systems using non-ASCII locales. After version 0.3, the filename is in a `uri` element, as a UTF-8 encoded URI, and the `filename` element is deleted if it is seen.

The result is this: opening files from 0.3 and earlier in a later version of Goanna works just fine without loss of data. However, if the file is saved, and the resulting file is re-opened in 0.3, the filename for the file source will be lost. If it is re-set, this change will be picked back up when it is re-opened in a latter version of Goanna.

Glossary

Dynamic property

A property of a node or other object that changes frequently in the course of operation of a system.

Node

A computer, network device (e.g. Infiniband switch), or other component which is a part of a system.

Scalable Systems Software (SSS)

A SciDAC (http://www.scidac.org) project to create a suite of management software for scalable high performance computing systems. More information may be found at the Scalable Systems Software web site (http://www.scidac.org/ScalableSystems/).

Static property

A property of a node or other object which is considered relatively static, assumed to last the lifetime of the object.

System

A cluster, supercomputer, network, or other object that is the subject of a Goanna file.